

## CLAIMS

What is claimed is:

*Sub 6*

1. A computing system, comprising:  
a rounding apparatus to accept an input value that is a real number  
3 represented in floating-point format, and to perform a rounding operation on the  
4 input value to generate an output value that is an integer represented in floating-  
5 point format;  
6 a memory to store a computer program that utilizes the rounding  
7 apparatus; and  
8 a central processing unit (CPU) to execute the computer program, the  
9 CPU is cooperatively connected to the rounding apparatus and the memory.
- 1 2. The system of claim 1, wherein the rounding apparatus uses a  
truncation technique to round the input value.
- 1 3. The system of claim 2, wherein the rounding apparatus includes:  
a floating-point to integer converter to truncate the input value to convert  
2 the input value to an integer represented in an integer format; and  
3 an integer to floating-point converter to convert the integer represented in  
4 an integer format to the output value.
- 1 4. The system of claim 1, wherein the rounding apparatus rounds the  
2 input value to the nearest integer.
- 1 5. The system of claim 4, wherein the rounding apparatus includes:  
2 an "AND" operator to extract a sign bit of the input value;  
3 an "OR" operator to generate an adjustment value based on the sign bit;

4 an ADD operator to compute an adjusted input value by adding the  
5 adjustment value to the input value, the adjusted input value is a real number  
6 represented in floating-point format;  
7 a floating-point to integer converter to truncate a fractional portion of the  
8 adjusted input value to convert the adjusted input value to an integer  
9 represented in an integer format; and  
10 an integer to floating-point converter to convert the integer represented in  
11 an integer format to generate the output value.

1 6. The system of claim 5, wherein the "AND" operator extracts the  
2 sign bit of the input value by performing a bit-wise logical AND operation on the  
3 input value and a sign mask.

1 7. The system of claim 5, wherein the "OR" operator generates the  
2 adjustment value by performing a bit-wise logical OR operation on the sign bit  
3 and a real value of 0.5.

1 8. The system of claim 1, wherein the rounding apparatus rounds the  
2 input value toward minus infinity ( $-\infty$ ).

1 9. The system of claim 8, wherein the rounding apparatus includes:  
2 a floating-point to integer converter to truncate an input value to convert  
3 the input value to a first integer represented in an integer format;  
4 an integer to floating-point converter to convert the first integer  
5 represented in an integer format to a second integer represented in floating-point  
6 format;  
7 a first SUBTRACT operator to compute a fractional portion of the input  
8 value using the second integer;

DRAFT 5/22/2019

10 a "less than" comparator to generate a boolean mask based on the  
11 fractional portion of the input value;  
12 an "AND" operator to use the boolean mask to generate an adjustment  
13 value represented in floating-point format; and  
14 a second SUBTRACT operator to subtract the adjustment value from the  
input value to generate the output value.

1 10. The system of claim 9, wherein the first SUBTRACT operator  
2 computes the fractional portion of the input value by subtracting the second  
3 integer from the input value.

1 11. The system of claim 9, wherein the "less than" comparator  
2 generates the boolean mask by comparing the fractional portion of the input  
3 value to a real value of 0.0.

1 12. The system of claim 9, wherein the "AND" operator generates the  
2 adjustment value by performing a bit-wise logical AND operation on the boolean  
3 mask and a real value of 1.0.

1 13. The system of claim 1, wherein the rounding apparatus rounds the  
2 input value toward plus infinity ( $+\infty$ ).

1 14. The system of claim 13, wherein the rounding apparatus includes:  
2 a floating-point to integer converter to truncate an input value to convert  
3 the input value to a first integer represented in an integer format;  
4 an integer to floating-point converter to convert the first integer  
5 represented in an integer format to a second integer represented in floating-point  
6 format;

*sub A*

8 a SUBTRACT operator to compute a fractional portion of the input value  
8 using the second integer;

9 a "greater-than" comparator to generate a boolean mask based on the  
10 fractional portion of the input value;

11 an "AND" operator to use the boolean mask to generate an adjustment  
12 value represented in floating-point format; and

13 an ADD operator to add the adjustment value to the input value to  
14 generate the output value.

1 15. The system of claim 14, wherein the SUBTRACT operator computes  
2 the fractional portion of the input value by subtracting the second integer from  
3 the input value.

1 16. The system of claim 14, wherein the "greater-than" comparator  
2 generates the boolean mask by comparing the fractional portion of the input  
3 value to a real value of 0.0.

1 17. The system of claim 14, wherein the "AND" operator generates the  
2 adjustment value by performing a bit-wise logical AND operation on the boolean  
3 mask and a real value of 1.0.

1 18. A method comprising:  
2 accepting an input value that is a real number represented in floating-  
3 point format;  
4 converting the input value to a first integer;  
5 converting the first integer represented to a second integer; and  
6 storing the second integer as an output value.

1           19. The method of claim 18, wherein converting the input value to a  
2           first integer comprises:  
3           representing the first integer in an integer format.

1           20. The method of claim 18, wherein converting the first integer to the  
2           second integer comprises:  
3           representing the second integer in floating-point format.

1           21. A method comprising:  
2           building an adjustment value represented in floating-point format;  
3           adding the adjustment value to an input value to generate an adjusted  
4           input value represented in floating-point format;  
5           truncating the adjusted input value to convert the adjusted input value to  
6           a first integer represented in an integer format;  
7           converting the first integer represented in an integer format to a second  
8           integer represented in floating-point format; and  
9           storing the second integer as an output value.

1           22. The method of claim 21, wherein building the adjustment value  
2           comprises:  
3           extracting a sign bit of the input value by performing a bit-wise logical  
4           AND operation on the input value and a sign mask.

1           23. The method of claim 21, wherein building the adjustment value  
2           comprises:  
3           building the adjustment value by performing a bit-wise logical OR  
4           operation on a real value of 0.5 and a sign bit extracted from the input value.

*1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11.*

24. A method comprising:  
1 generating a first integer represented in an integer format by truncating  
2 an input value;  
3 converting the first integer represented in an integer format to a second  
4 integer represented in floating-point format;  
5 computing a fractional portion of the input value using the second integer  
6 represented in floating-point format;  
7 generating a boolean value using the fractional portion of the input value;  
8 creating an adjustment value using the boolean value;  
9 computing a rounded input value by subtracting the adjustment value  
10 from the input value.

1 25. The method of claim 24, wherein computing the fractional portion  
2 of the input value comprises:  
3 subtracting the second integer represented in floating-point format from  
4 the input value to generate the fractional portion of the input value.

1 26. The method of claim 24, wherein generating the boolean value  
2 comprises comparing the fractional portion of the input value to a real value of  
3 0.0.

1 27. The method of claim 24, wherein creating an adjustment value  
2 comprises performing a bit-wise logical AND operation on the boolean value  
3 and a real value of 1.0.

1 28. A method comprising:  
2 generating a first integer represented in an integer format by truncating  
3 an input value;

4 *Sub A* converting the first integer represented in an integer format to a second  
5 integer represented in floating-point format;  
6 subtracting the second integer represented in floating-point format from  
7 the input value to generate a fractional portion of the input value;  
8 generating a boolean value using the fractional portion of the input value;  
9 creating an adjustment value using the boolean value;  
10 adding the adjustment value to the input value to generate a rounded  
11 input value.

1 29. The method of claim 28, wherein creating an adjustment value  
2 comprises:  
3 comparing the fractional portion of the input value to a real value of 0.0.

1 30. The method of claim 28, wherein creating an adjustment value  
2 comprises:  
3 performing a bit-wise logical AND operation on the boolean value and a  
4 real value of 1.0.

1 31. A machine-readable medium comprising instructions which, when  
2 executed by a machine, cause the machine to perform operations comprising:  
3 a first code segment truncates the input value to convert the input value to  
4 a first integer; and  
5 a second code segment integer to convert the first integer to a second  
6 integer.

1 32. The machine-readable medium of claim 31, wherein the first integer  
2 is represented in an integer format.

*Claim 33*  
33. The machine-readable medium of claim 31, wherein the second integer is represented in floating-point format.

1           34. A machine-readable medium comprising instructions which, when  
2   executed by a machine, cause the machine to perform operations comprising:  
3        a first code segment to extract a sign bit of the input value;  
4        a second code segment to generate an adjustment value based on the sign  
5   bit;  
6        a third code segment to compute an adjusted input value represented in  
7   floating-point format;  
8        a fourth code segment to truncate a fractional portion of the adjusted  
9   input value to convert the adjusted input value to an integer represented in an  
10   integer format; and  
11        a fifth code segment to convert the integer represented in an integer  
12   format to generate the output value.

1           35. The machine-readable medium of claim 34, wherein the second  
2   code segment generates the adjustment value by performing a bit-wise logical  
3   OR operation on the sign bit and a value of 0.5.

1           36. The machine-readable medium of claim 34, wherein the third code  
2   segment computes the adjusted input value by adding the adjustment value to  
3   the input value.

1           37. A machine-readable medium comprising instructions which, when  
2   executed by a machine, cause the machine to perform operations comprising:  
3        a first code segment to truncate an input value to convert the input value  
4   to a first integer represented in an integer format;

5           a second code segment to floating-point converter to convert the first  
6 integer represented in an integer format to a second integer represented in  
7 floating-point format;  
8           a third code segment to subtract the second integer from the input value  
9 to compute a fractional portion;  
10          a fourth code segment to generate a boolean mask based on the fractional  
11 portion of the input value;  
12          a fifth code segment to perform a bit-wise logical AND operation on the  
13 boolean mask and a real value of 1.0 to generate an adjustment value represented  
14 in floating-point format; and  
15          a sixth code segment to subtract the adjustment value from the input  
16 value to generate the output value represented in floating-point format.

1           38. The machine-readable medium of claim 37, wherein the fourth code  
2 segment generates the boolean mask by comparing the fractional portion of the  
3 input value to a real value of 0.0.

1           39. The machine-readable medium of claim 37, wherein the fifth code  
2 segment generates the adjustment value by performing a bit-wise logical AND  
3 operation on the boolean mask and a real value of 1.0.

2           40. A machine-readable medium comprising instructions which, when  
3 executed by a machine, cause the machine to perform operations comprising:  
4           a first code segment to truncate an input value to convert the input value  
5 to a first integer represented in an integer format;  
6           a second code segment to floating-point converter to convert the first  
7 integer represented in an integer format to a second integer represented in  
floating-point format;

*del pat*

8 a third code segment to subtract the second integer from the input value  
9 to compute a fractional portion of the input value;  
10 a fourth code segment to generate a boolean mask based on the fractional  
11 portion of the input value;  
12 a fifth code segment to an adjustment value represented in floating-point  
13 format; and  
14 a sixth code segment to subtract the adjustment value from the input  
15 value to generate the output value represented in floating-point format.

1 41. The machine-readable medium of claim 40, wherein the fourth code  
2 segment generates the boolean mask by comparing the fractional portion of the  
3 input value to a real value of 0.0.

1 42. The machine-readable medium of claim 40, wherein the fifth code  
2 segment generates the adjustment value by performing a bit-wise logical AND  
3 operation on the boolean mask and a real value of 1.0.